

J A V A P r o j e c t

# JAVA 프로젝트 대여 관리 시스템

[ 조 원 ]

남궁현, 이현호, 권용민,  
전성현, 박명지

# CONTENTS

01

트

02

트

03

상 관  
계 도  
속

04

기 능  
소 개

05

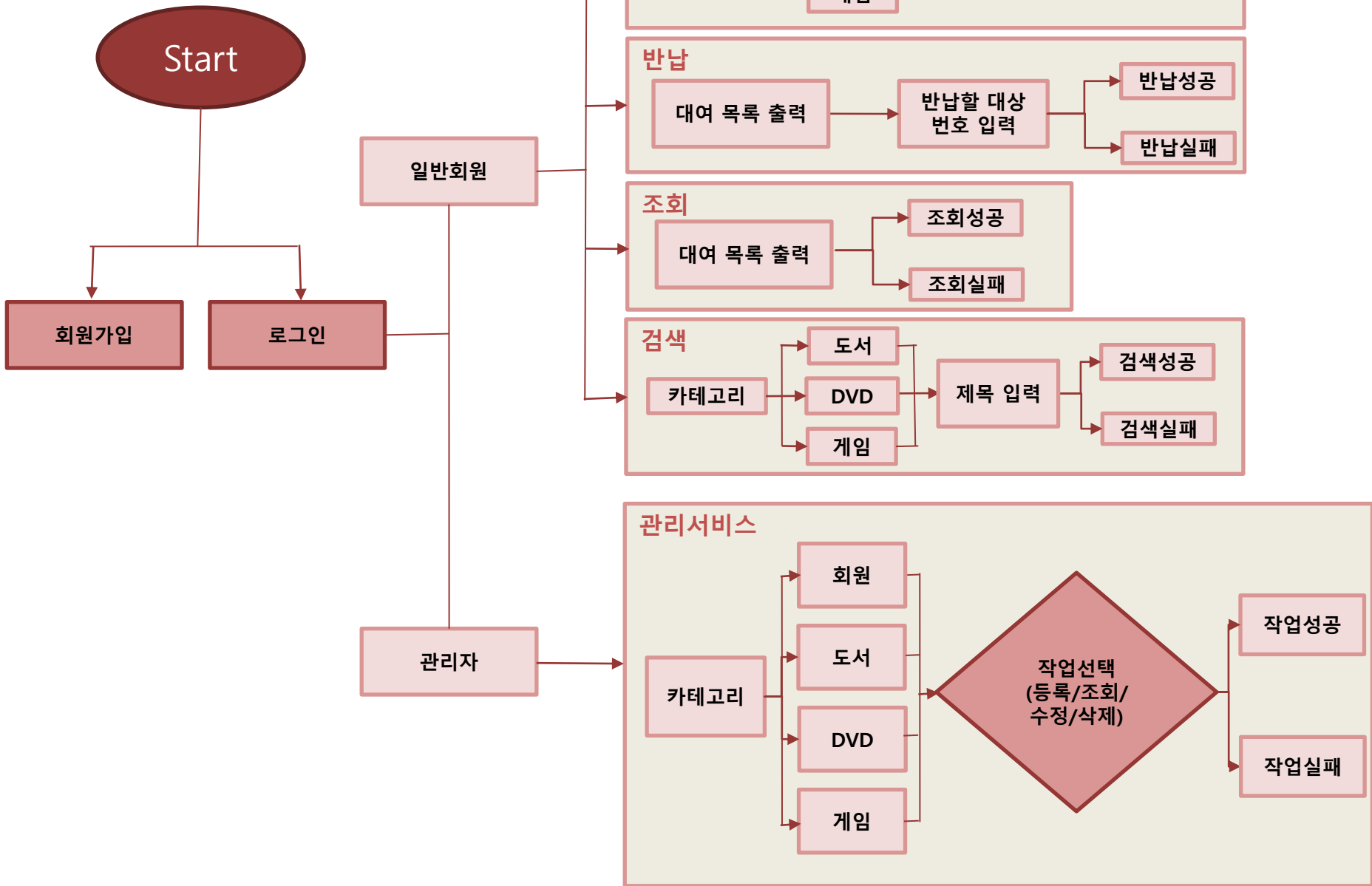
문제해결  
및  
개선사항

## “대여관리 시스템”

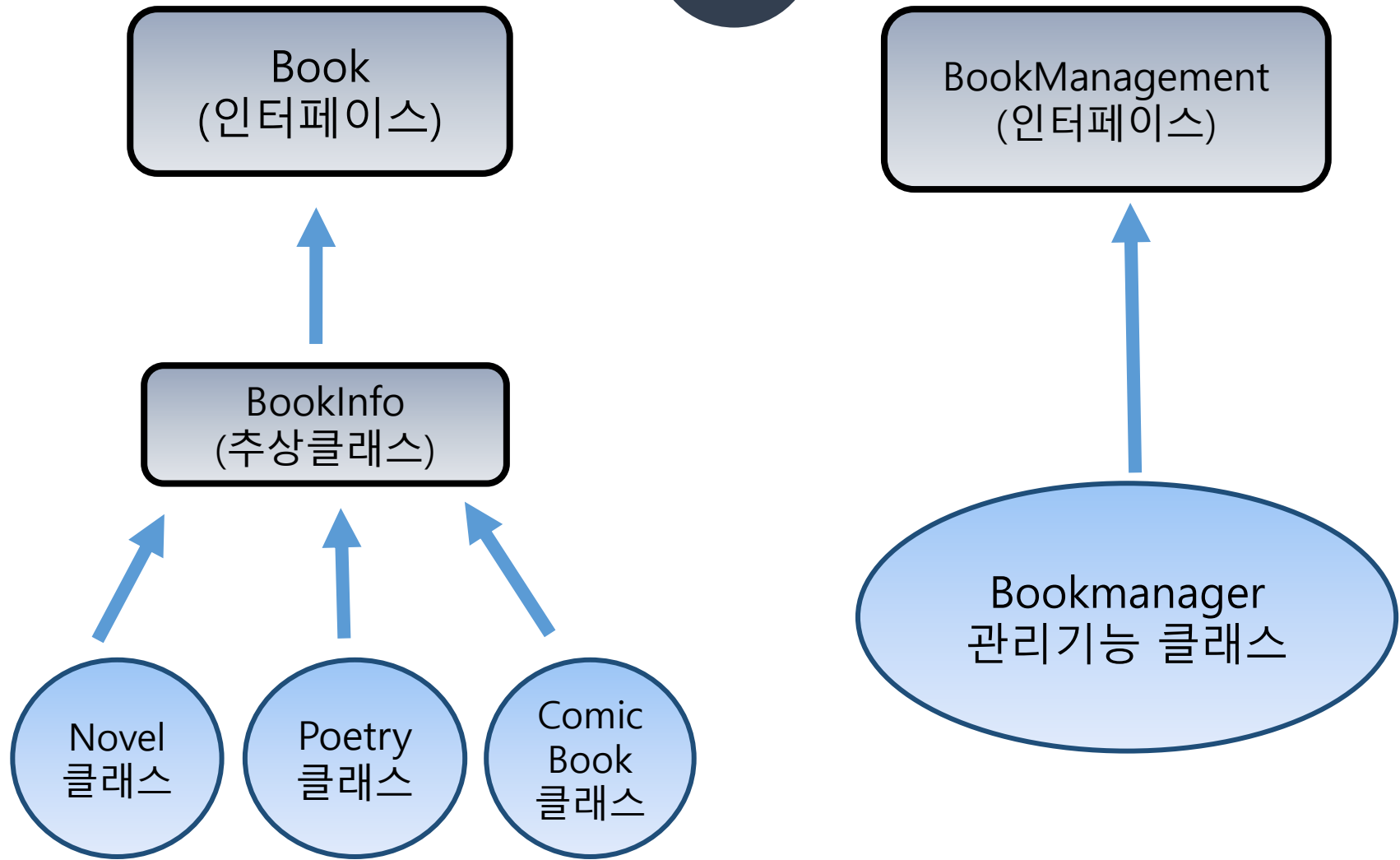
간단한 인증 절차를 통해 회원들이 도서, DVD, 게임 등을  
대여/조회/검색 등 서비스를 자유롭게 이용 할 수 있으며,

관리 메뉴 또한 관리자가 보다 쉽게 관리할 수  
있도록 간단한 메뉴들로 구성

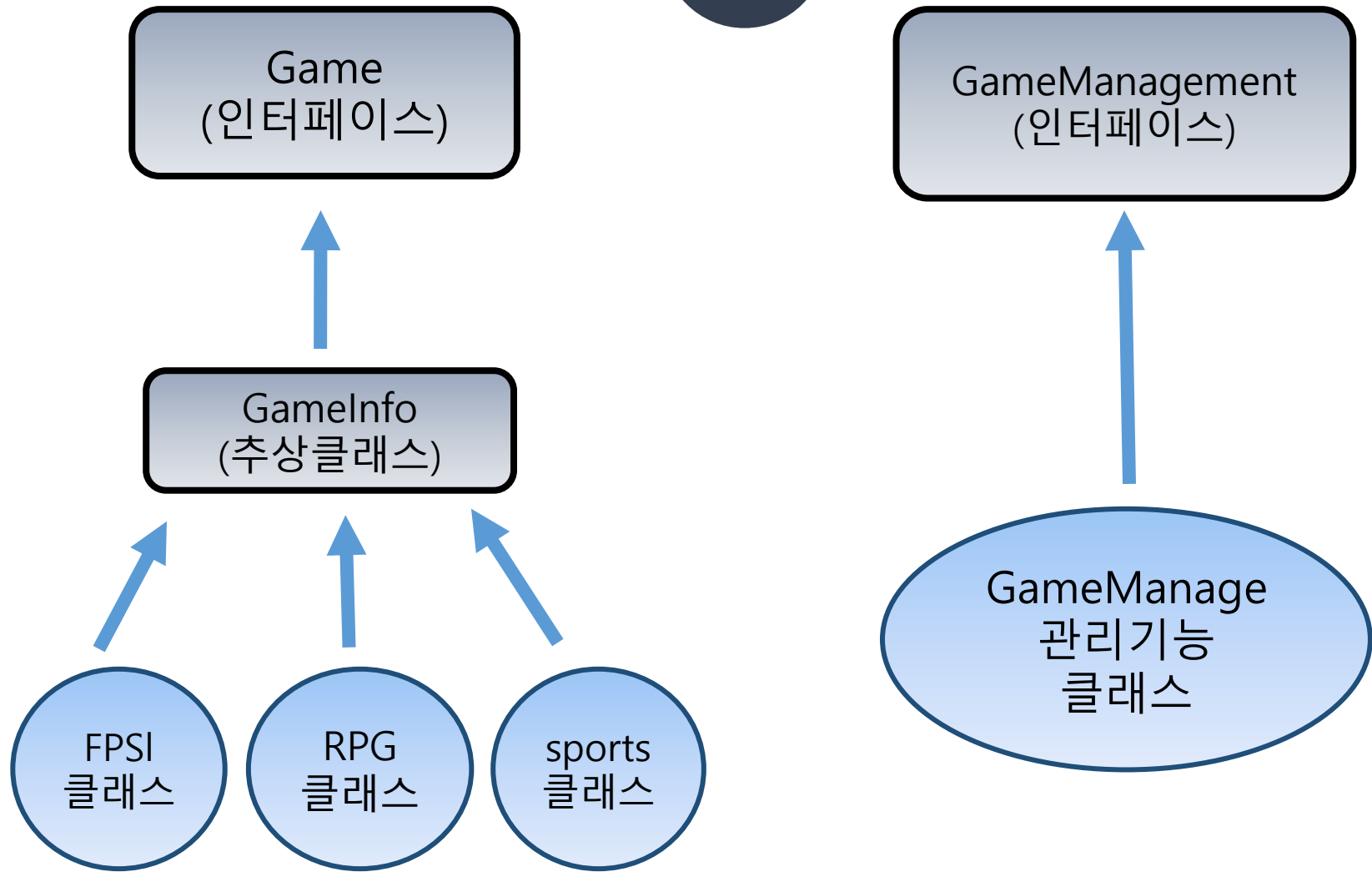
## 2. 프로젝트 흐름도



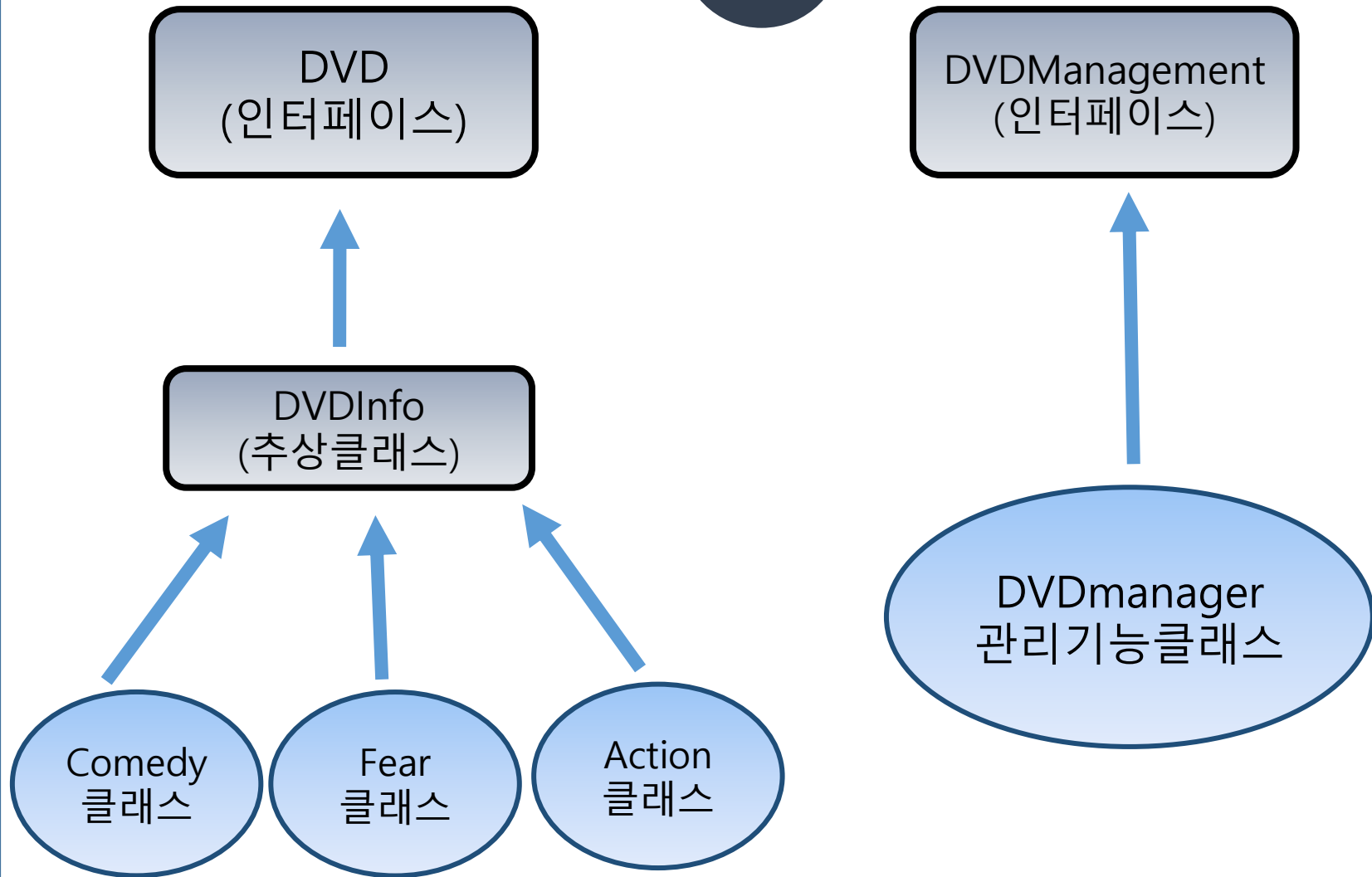
### 3. 상속 관계도



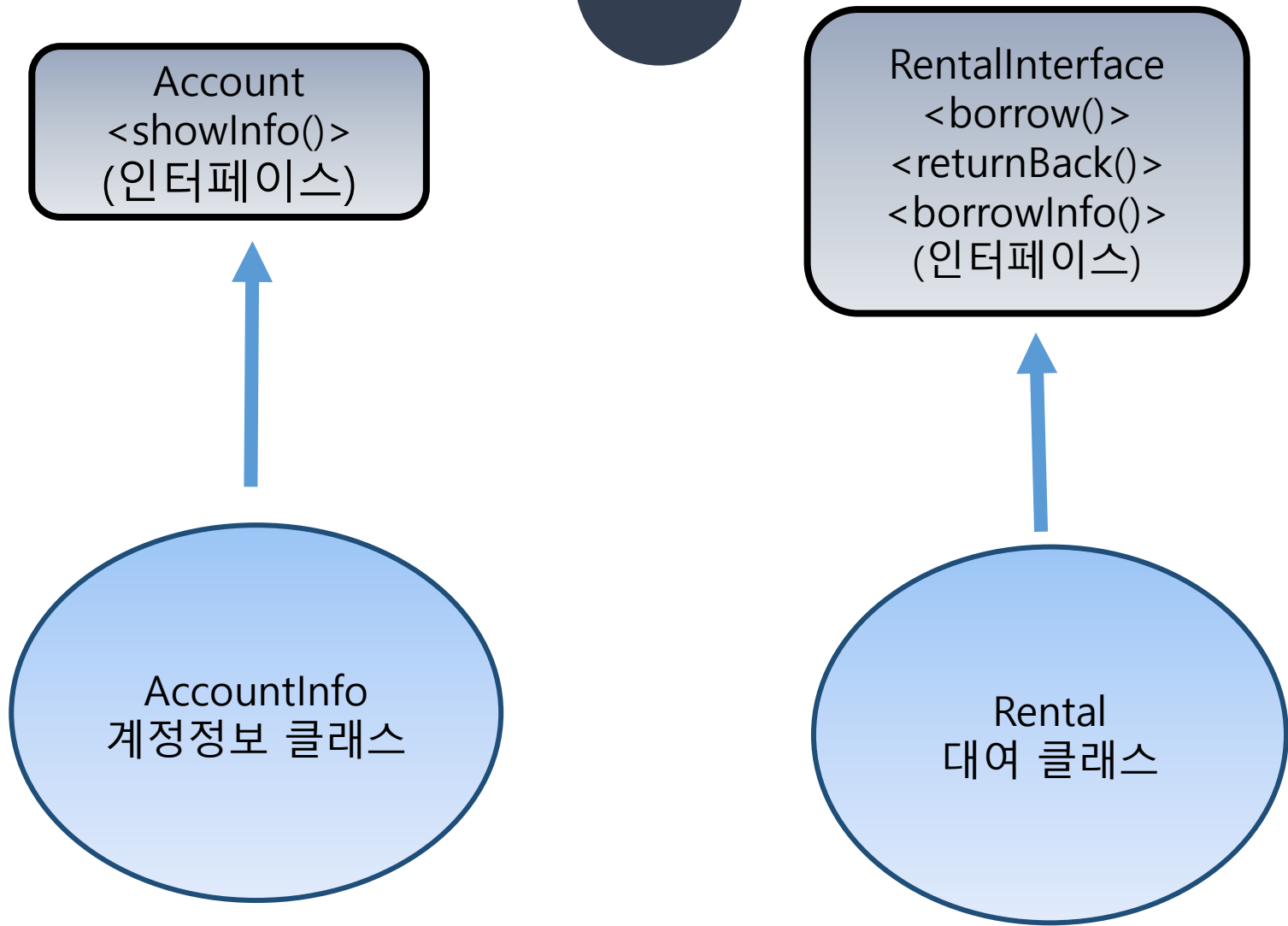
### 3. 상속 관계도



### 3. 상속 관계도



### 3. 상속 관계도





## 4. 기 능 소 개

# 회 원 관 리

```
// 메인 화면
public int insertMenu() {

    System.out.println("=====");
    System.out.println("메뉴를 선택하세요");
    System.out.printf("%d.로그인 \n%d.회원가입 \n",Util.LOGIN,Util.JOIN);
    System.out.println("=====");

    int keyboard = Util.inputInt();

    while(keyboard<1 || keyboard >2){
        System.out.println("1번에서 2번 사이의 값을 입력하세요.");
        keyboard = Util.inputInt();
    }

    return keyboard;
}

// 관리자 메뉴 화면
int detailMenu() {

    System.out.println("=====");
    System.out.println("이용하실 메뉴를 선택하세요");
    System.out.printf("%d.회원생성 \n%d.회원검색 \n%d.회원삭제 \n%d.회원수정 \n%d.모든 회원검색 \n%d.종료 \n",
        Util.INSERT,Util.SEARCH,Util.DELETE,Util.MODIFY,Util.SEARCH_ALL,Util.QUIT2);
    System.out.println("=====");

    int keyboard = Util.inputInt();

    return keyboard;
}
```

## <메뉴 화면>

- insertMenu 메서드는  
최초 프로그램 실행시에  
로그인, 회원가입 메뉴 구현
- detailMenu 메서드는  
관리자 계정으로 접속 시  
회원관리 메뉴 구현

```
public class AccountValidateCheck {  
  
    // ID 일치여부 확인  
    AccountInfo IdCheck(String id) {  
        AccountManager manager = AccountManager.getInstance();  
  
        for(AccountInfo idcheck : manager.getMembers()) {  
            if(idcheck.getID().equals(id)) {  
                return idcheck;  
            }  
        }  
        return null;  
    }  
  
    // ID 조합 규칙 메서드  
    int checkID(String id) {  
        int isCheck = 1;  
        if(id.length() < 5) {  
            isCheck = 0;  
        }  
        return isCheck;  
    }  
  
    // PW 조합 규칙 메서드  
    int checkPassword(String password) {  
        int isCheck = 1;  
        char[] ch = password.toCharArray();  
  
        // 패스워드 9자 이상  
        if (password.length() >= 9) {  
            // 패스워드 대문자 포함여부 확인  
            for(int i=0; i<ch.length; i++) {  
                if(ch[i] >= 65 && ch[i] <= 90) {  
                    isCheck = 2;  
                }  
            }  
            // 패스워드 특수문자 포함여부 확인  
            for(int i=0; i<ch.length; i++) {  
                if (ch[i] >= 33 && ch[i] <= 47) {  
                    isCheck = 3;  
                }  
            }  
        } else {  
            isCheck = 1;  
        }  
        return isCheck;  
    }  
}
```

## <유효성 체크>

- 로그인 및 회원가입 시  
ID 및 Password 유효성 검사
- checkID로 ID 길이 제어  
/ checkPW로 PW 길이 제어  
및 영어 대문자+특수문자  
포함 여부 검사
- IdCheck 메서드는 ArrayList  
객체를 참조타입의 idcheck  
에 대입하여 getID() 값과  
입력한 id 값을 비교(equals)

```
void createAccount() {  
  
    Util.keyboard.nextLine();  
  
    String id = null;  
    String password = null;  
    String name = null;  
    String phoneNumber = null;  
  
    while(true) {  
        System.out.println("회원님의 정보를 입력해주세요.");  
        System.out.print("ID :");  
        id = Util.keyboard.nextLine();  
  
        // ID 중복 체크  
        if(vaild.IdCheck(id) != null){  
            System.out.println("이미 등록된 ID 입니다.");  
            continue;  
        }  
        // ID 길이 체크  
        else if(vaild.checkID(id) == 0) {  
            System.out.println("5자 이상 입력해주세요.");  
            continue;  
        }  
  
        System.out.print("PW :");  
        password = Util.keyboard.nextLine();  
  
        // PW 길이 체크  
        if(vaild.checkPassword(password) == 1) {  
            System.out.println("9자 이상 입력해주세요.");  
            continue;  
        }  
  
        // PW 조합 체크  
        else if(vaild.checkPassword(password) != 3) {  
            System.out.println("특수문자 혹은 영어 대문자를 포함해 주세요.");  
            continue;  
        }  
  
        System.out.print("NAME :");  
        name = Util.keyboard.nextLine();  
  
        System.out.print("Phone :");  
        phoneNumber = Util.keyboard.nextLine();  
  
        System.out.println("회원가입이 완료 되었습니다.");  
        //사용자로부터 입력받은 데이터로 인스턴스 생성 (객체의 변수에 저장)  
        members.add(new AccountInfo(id, password, name, phoneNumber));  
  
        break;  
    }  
}
```

## <회원가입 기능>

- id 입력 시 IdCheck 메소드로 등록된 사용자인지 검사
- id / password 입력 시 checkID / checkPW 메소드 조합 규칙 검사
- 입력한 필드들로 member ArrayList 에 저장

```
// 로그인 메서드
void accountlogin() {
    //현재 입력 스캐너의 버퍼를 삭제
    Util.keyboard.nextLine();

    String id = null;
    String password = null;

    System.out.println("로그인 정보를 입력해주세요.");

    while(true) {

        System.out.print("ID :");
        id = Util.keyboard.nextLine();

        System.out.print("PW :");
        password = Util.keyboard.nextLine();

        // ID 일치여부 확인
        if(vaild.IdCheck(id) == null){
            System.out.println("등록되지 않은 ID입니다.");
            continue;
        }
        // 저장된 PW 와 입력한 PW 비교
        }else if(vaild.IdCheck(id).getPW().equals(password)) {
            System.out.println("[ " + vaild.IdCheck(id).getID() + " ]님께서 로그인 하셨습니다.");
            loginId = id;
        }else {
            System.out.println("비밀번호가 틀렸습니다.");
            continue;
        }
    }
    break;
}
```

## <로그인 기능>

- id 입력 시 IdCheck 메소드로 등록된 사용자인지 검사
- password 입력 시 idCheck 객체의 password 값과 비교 (equals)
- loginId 에 id 를 저장하여 대여관리 사용

```
// 회원정보 삭제
void deleteAccount() {

    Util.keyboard.nextLine();
    System.out.println("삭제하실 회원님의 ID를 입력하세요. :");
    String id = Util.keyboard.nextLine();

    // ID 일치여부 확인
    if(vaild.IdCheck(id)==null) {
        System.out.println("등록되지 않은 ID입니다.");
    }

    for(int i=0; i<members.size(); i++) {
        if(members.get(i).getID().equals(id)) {
            members.remove(i);
            System.out.println("[삭제정보 안내]");
            System.out.println("삭제된 ID : "+ id);
            break;
        }
    }
}
```

## <회원삭제 기능>

- id 입력 시 IdCheck 메소드로 등록된 사용자인지 검사
- 입력한 id와 members의 id가 일치하면 해당 index 삭제
- 삭제 완료 후 id 출력

```
// 회원정보 검색
void searchAccount() {

    Util.keyboard.nextLine();
    System.out.println("검색하실 회원님의 ID를 입력하세요. ");
    String id = Util.keyboard.nextLine();

    AccountInfo search = vaild.IdCheck(id);

    // ID 일치여부 확인
    if(search == null){
        System.out.println("등록된 ID가 없습니다.");
    } else {
        System.out.println("[회원정보 안내]");
        System.out.println("ID : "+ search.getID());
        System.out.println("PW : "+ search.getPW());
        System.out.println("NAME : "+ search.getName());
        System.out.println("Phone : "+ search.getPhoneNumber());
    }
}

// 모든 회원 검색
void showAll() {
    for(AccountInfo info : members) {
        System.out.println("ID : "+info.getID()+"\tPW : "+info.getPW()+
            "\tNAME : "+info.getName()+"\tPhone : "+info.getPhoneNumber());
    }
}
```

## <회원검색 기능>

- id 입력 시 IdCheck 메소드로 등록된 사용자인지 검사
- search 변수에 getter 를 이용 저장된 객체의 정보를 출력
- showAll 메서드는 for-each문으로 members 에 저장된 객체를 모두 출력

## 4. 기 능 소 개

# 게 임 관 리



```

public int showMenu() {                                     // 메인메서드 시작시 시작메뉴 메서드
    System.out.println("메뉴를 선택해주세요.");
    System.out.println("=====");
    System.out.printf("%d.게임 등록\n%d.게임 삭제\n%d.게임 검색\n%d.전체 게임\n%d.종료"
        ,Util.INSERT,Util.DELET,Util.SERCH,Util.ALLLIST,Util.QUIT);
    System.out.println("");
    System.out.println("=====");

    inputException();                                       //예외처리
    keyboard.nextLine();

    return select;
}

```

게임 관리 메뉴입니다. 번호를 입력시 inputException (예외처리) 메서드를 거쳐 select 으로 리턴됩니다.

## Game 입력

```

public void insertGame() { // 삽입기
    System.out.println("게임 종류를 선택해주세요.");
    System.out.println("=====");

    for(int i = 1; i<=gameList.length; i++) {
        System.out.printf("%d. %s\n",i,gameList[i-1]);
    }

    inputException2(); // 예외
    keyboard.nextLine();

    System.out.println("=====");
    System.out.println("게임 이름을 입력해주세요.");
    String gameName = keyboard.nextLine();

    String type = gameList[select-1];

    String unumber = "GAME"+ type+ uniqueNumber++; //고유번호

    if(select == 1) {
        game.add(new Fps(unumber,gameName,type)); //rent
    }else if(select == 2) {
        game.add(new RPG(unumber,gameName,type));
    }else if(select == 3) {
        game.add(new Sport(unumber,gameName,type));
    }else {
        System.out.println("선택된 번호 안에서 골라주세요.");
    }
    System.out.println("게임이 저장되었습니다.");
    System.out.println("=====");
}

```

게임 입력기능 입니다.  
 게임 종류를 먼저 선택하여  
 InputException2 (예외처리)메  
 서드를 거쳐 select 으로 리턴  
 되어오고 gameName에 이름을  
 입력하고 게임 type(종류)은  
 종류배열(gameList) 로 입력  
 되고 선택한 번호(select)로  
 종류별 배열에 입력됩니다.

## Game 삭제

```

public void deletGame() {                                // 게
    searchGame();                                        // 검
    if(searchResult.size() < 1) {
        switchnum = 0;
        return;
    }
    System.out.println("삭제할 번호를 선택해주세요.");

    select = keyboard.nextInt();

    game.remove(searchResult.get(select-1));
    System.out.println("선택하신 게임이 삭제되었습니다.");
    System.out.println("1. 계속 삭제 2. 메뉴로 돌아가기");
    select = keyboard.nextInt();
    if(select == 1) {
        switchnum = 1;
        deletGame();
    }else if(select == 2) {
        System.out.println("종료합니다.");
        switchnum = 0;
    }
}

```

게임 삭제 기능 입니다.  
 검색기능을 거쳐 검색된 이름의  
 게임들을 가져와 번호로  
 나타냅니다. 그중에서 삭제할  
 번호를 입력하면 검색된게임  
 배열에서 선택한 번호의 게임이  
 삭제되고 계속 삭제할것인지  
 메인 메뉴로 돌아갈것인지  
 선택합니다.

## Game 검색

```

public void searchGame() {
    searchResult.clear();

    if(switchnum == 0) {
        System.out.println("게임이름을 검색해주세요.");
        gameName = keyboard.nextLine();
    }

    int k = 1;
    for(int i=0; i<game.size(); i++) {
        if(game.get(i).gameName.equals(gameName)) {
            System.out.println("====[ "+ k++ + " ]====");
            game.get(i).gameInfo();
            searchResult.add(game.get(i));
        }
    }
    if(searchResult.size() < 1) {
        System.out.println("검색하신 게임이 없습니다.");
        System.out.println("=====");
        return;
    }
}

```

게임 검색 기능 입니다.  
 Switchnum = 0 으로 초기화 되어있고 검색기능 시작시  
 gameName 에 게임이름을  
 입력하고 저장되어있는 총 게임  
 의 수만큼 0 부터 반복해서 그중  
 검색한 이름과 같은 이름의 게임을  
 찾아 정보를 보여주고  
 SearchResult 라는 배열에 입력  
 시켜준다.  
 SearchResult 배열이 1보다작다  
 면 검색한 게임이 없다고 나온다.

```
public void allListGame() { // 전체 정보 출력
    for(int i=0; i<game.size(); i++) {

        System.out.println("====[ "+ (i+1) +" ]====");
        game.get(i).gameInfo();
    }
    if(game.size()<1) {
        System.out.println("저장되어있는 게임이 없습니다.");
        System.out.println("=====");
        return;
    }
}
```

저장되어있는 게임전체를 출력해주는 기능입니다.  
저장되어있는 게임의 수만큼 0부터 반복시키면서 게임의 정보를 출력해줍니다. 만약 게임배열의 크기가 1보다 작을시 저장되어있는 게임이 없으므로 리턴 시켜 종료시켜줍니다.

```
public void inputException() {  
    try {  
        select = keyboard.nextInt();  
    } catch (InputMismatchException e) {  
        System.out.println("잘못입력하셨습니다. 다시입력하세요.");  
        select = 0;  
        return;  
    }  
}
```

예외처리 기능을하는 메서드입니다.

Select 에 Int 타입이 들어가지않으면 select 를 0으로 초기화시켜주고 리턴 합니다.

## 4. 기 능 소 개

# 도서 관 리

## <초기메뉴 화면>

Book관리 - BookManager

```
public int showMenu() { // 메뉴 메서드
    System.out.println("메뉴를 선택해주세요.");
    System.out.println("-----");
    System.out.printf("%d.책 등록\n%d.책 삭제\n%d.책 검색\n%d.전체 책목록\n%d.종료", Util.INSERT, Util.DELET, Util.SERCH,
        Util.ALLLIST, Util.QUIT);
    System.out.println("");
    System.out.println("-----");

    exception1();
    keyboard.nextLine();

    return select;
}
```

- 프로그램을 실행하면 제일 먼저 출력되는 메서드입니다.
- 메뉴 목록에는 등록, 삭제, 검색, 전체 책목록, 종료가 있습니다.
- 예외처리 후 해당하는 메뉴가 실행됩니다.



# <저장 기능>

Book관리 - BookManager

```
@Override
public void insertBook() { // 책등록 메서드
    System.out.println("책 종류를 선택해주세요");
    System.out.println("-----");

    for (int i = 1; i <= bookList.length; i++) {
        System.out.printf("%d. %s\n", i, bookList[i - 1]);
    }
    exception2();
    keyboard.nextLine();

    System.out.println("-----");
    System.out.println("책 제목을 입력해주세요.");
    String bookName = keyboard.nextLine();
    System.out.println("-----");
    System.out.println("작가를 입력해주세요.");
    String writer = keyboard.nextLine();
    System.out.println("-----");
    System.out.println("평점을 입력해주세요.(1~5)");
    int grade = keyboard.nextInt();

    String type = bookList[select - 1];

    if (select == 1) {
        book.add(new Novel(bookName, writer, type, grade));
    } else if (select == 2) {
        book.add(new Poetry(bookName, writer, type, grade));
    } else if (select == 3) {
        book.add(new ComicBook(bookName, writer, type, grade));
    } else {
        System.out.println("잘못 입력하셨습니다. 다시 입력해주세요");
    }

    System.out.println("책정보가 저장되었습니다.");
    System.out.println("-----");
}
```

- 책의 정보를 저장하는 기능의 메서드입니다.
- 원하는 종류(장르)의 번호를 입력합니다.
- 예외처리(Exception2) 후 select를 통해 다시 해당 입력값으로 돌아와 제목,작가명,장르,평점을 입력하면 저장이 완료됩니다.

```
@Override
public void deleteBook() {
    searchBook();
    if (searchResult.size() < 1) {
        switchNum = 0;
        return;
    }
    System.out.println("삭제할 번호를 선택해주세요");
    int select = keyboard.nextInt();

    book.remove(searchResult.get(select - 1));
    System.out.println("선택하신 책이 삭제되었습니다.");
    System.out.println("1.계속삭제 2. 종료");
    select = keyboard.nextInt();

    if (select == 1) {
        switchNum = 1;
        deleteBook();
    } else if (select == 2) {
        System.out.println("종료합니다");
        switchNum = 0;
    }
}
```

저장되어있는 책의정보를 삭제하는 기능을 구현한 메서드 입니다.

삭제하고자 하는 책의 제목을 입력하면 검색메서드를 통해 등록되어 있는 책의 목록과 해당 번호가 출력됩니다.

해당번호를 입력하면 삭제가 완료되고 계속삭제를 원한다면 해당 번호를 입력, 삭제메서드가 다시 실행되고 종료를 하면 실행할게 없으므로 다시 초기 메뉴화면으로 돌아가게 된다.

```
@Override
public void searchBook() {
    searchResult.clear();

    if (switchNum == 0) {
        System.out.println("책 제목을 검색해주세요.");
        bookName = keyboard.nextLine();
    }

    int k = 1;
    for (int i = 0; i < book.size(); i++) {
        if (book.get(i).bookName.equals(bookName)) {
            System.out.println("====[ " + k++ + " ]====");
            book.get(i).bookInfo();
            searchResult.add(book.get(i));
        }
    }
    if (searchResult.size() < 1) {
        System.out.println("검색하신 책을 찾을 수 없습니다.");
        System.out.println("-----");
        return;
    }
}
```

- 현재 저장되어 있는 책을 검색 할 수 있는 메서드 입니다.
- 저장되어있던 검색결과를 clear로 검색 공간을 비우고 찾고자하는 책의 제목을 검색합니다.
- book배열에 저장되어 있는 객체에 저장되어 있는 책이름과 입력한 책 제목을 비교 후 같으면 리스트로 출력합니다.

```
@Override
public void allListBook() {
    for (int i = 0; i < book.size(); i++) {

        System.out.println("====[" + (i + 1) + "]====");
        book.get(i).bookInfo();
    }
}
}
```

등록되어 있는 책 전체의 목록을 보여주는 메서드입니다.  
반복문을 사용하여 i를 size만큼 증가시키고 해당 범위까지 등록되어 있는 책의 정보들을 출력하게 됩니다.

```
public void exception1() { // 메뉴 예외처리 메서드
    try {
        select = keyboard.nextInt();
    } catch (Exception e) {
        System.out.println("잘못입력하셨습니다. 다시 입력하세요");
        select = 0;
        return;
    }
}
```

- select를 통해 int가 입력되면 정상적으로 실행 됩니다.
- Int가 아닌 이외의 것이 입력되면  
select에 입력된 값이 0으로 초기화가 이루어지며 return 됩니다.

## 4. 기 능 소 개

# DVD 관 리

## DVDInfo 클래스

```
public abstract class DvdInfo implements Dvd {  
  
    String dName;  
    String dUnique;  
    String rental;  
    int grade;  
  
    public DvdInfo(String dName, String dUnique, int grade) {  
        this.dName = dName;  
        this.dUnique = dUnique;  
        this.grade = grade;  
        this.rental = "대여가능";  
    }  
  
    public void showData() {  
        System.out.println("제목: " + dName);  
        System.out.println("고유번호: " + dUnique);  
        System.out.println("대여여부: " + rental + "합니다.");  
        System.out.print("평점은:");  
        for (int i = 1; i <= grade; i++) {  
            System.out.print("★");  
        }  
        System.out.println();  
    }  
}
```

DvdInfo 클래스 입니다.  
Dvd 인터페이스를  
상속하고 구현합니다.  
Dvd이름과 고유번호,  
평점을 매개 변수로,  
평점은 입력받는 숫자에  
맞게 저장되도록 했습니  
다.

## 04

## 메뉴 메서드

```
public int printMenu() { // 메뉴 메서드
    System.out.println("=====선택=====");
    System.out.println("메뉴를 선택해 주세요.");
    System.out.printf("%d.dvd등록%n%d.dvd삭제%n%d.dvd검색%n%d.전체정보%n%d.종료%n", Menu.INSERT, Menu.DELET, Menu.SERCH,
        Menu.ALLLIST, Menu.QUIT);
    System.out.println("=====");
    // 예외처리 메서드호출
    inputException();
    choice.nextLine();
    return select;
}
```

dvd관리 메뉴 입니다.  
예외처리 메서드를 거쳐 번호입력시 해당번호를  
int 로 반환합니다.



## 예외처리 메서드

```
private void inputException() { // 메뉴 예외처리 메서드
    try {
        select = choice.nextInt();
    } catch (InputMismatchException e) {
        System.out.println("잘못입력하셨습니다. 다시입력하세요.");
        select = 0;
        return;
    }
}
```

Try 블록 내에서 예외가 발생하면 다음 catch의 문장을 실행하고 select에 0을 대입한 후 리턴합니다.

## 등록메서드

```
@Override
public void insertDvd() {
    System.out.println("dvd 종류를 선택해주세요.");
    System.out.println("=====");
    for (int i = 1; i <= dvdList.length; i++) {
        System.out.printf("%d. %s\n", i, dvdList[i - 1]);
    }
    inputException2();

    choice.nextLine();
    System.out.println("=====");
    System.out.println("dvd이름을 입력해주세요.");
    String dName = choice.nextLine();

    System.out.println("평점을 입력하세요(1~5점)");
    int grade = choice.nextInt();

    String type = dvdList[select - 1];

    String uNum = "DVD" + type + uniqueNum++;
    if (select == 1) {
        dvd.add(new DvdAction(dName, uNum, type, grade));
    } else if (select == 2) {
        dvd.add(new DvdComedy(dName, uNum, type, grade));
    } else if (select == 3) {
        dvd.add(new DvdFear(dName, uNum, type, grade));
    }
    System.out.println("dvd가 저장되었습니다.");
    System.out.println("=====");
}
```

dvd등록시 장르를  
선택합니다.  
예외처리 거쳐 숫자  
입력시 해당 장르->이름  
-> 평점을 입력합니다.  
각 입력하신 정보를  
장르와 함께 고유번호를  
매겨 저장합니다.

## 검색 메서드

```
@Override
public void searchDvd() { // 검색 메서드
    sr.clear();

    if (swNum == 0) {
        System.out.println("dvd이름을 검색해주세요");
        dvdName = choice.nextLine();
    }
    int j = 1;
    for (int i = 0; i < dvd.size(); i++) {
        if (dvd.get(i).dName.equals(dvdName)) {
            System.out.println("=====[ " + j++ + " ]=====");
            dvd.get(i).showData();
            sr.add(dvd.get(i));
        }
    }
    if (sr.size() < 1) {
        System.out.println("검색하신 dvd가 없습니다.");
        System.out.println("=====");
        return;
    }
}
```

Dvd 검색 기능  
Dvd 이름 입력하면 같은  
이름이 나올때까지  
반복해서 찾아 검색한  
이름과 같은게 있으면  
정보를 보여주고 배열에  
입력합니다.

배열에 입력되어 있는게  
없으면 검색결과가 없다고  
나옵니다.

## 삭제 메서드

```
@Override
public void deletDvd() { // 삭제 메서드
    searchDvd();
    if (sr.size() < 1) {
        swNum = 0;
        return;
    }
    System.out.println("삭제할 번호를 선택해주세요.");
    int select = choice.nextInt();

    dvd.remove(sr.get(select - 1));
    System.out.println("선택하신 dvd가 삭제되었습니다.");
    System.out.println("1. 계속삭제  2. 종료(메뉴) ");
    select = choice.nextInt();
    if (select == 1) {
        swNum = 1;
        deletDvd();
    } else if (select == 2) {
        System.out.println("종료합니다.");
        swNum = 0;
    }
}
```

Dvd 삭제를 선택하면  
검색 메서드가 실행되어  
삭제하고자 하는 이름을  
입력합니다.  
검색한 결과가 없으면  
다시 메뉴화면으로  
이동하고 삭제할 이름과  
같은게 있으면 등록된  
정보와 번호가 출력됩니다.

삭제 하고싶은 번호를  
입력시 삭제 한뒤  
계속 삭제 또는 종료를  
할수 있습니다.

## 전체정보 메서드

```
@Override
public void allListDvd() { //전체정보 메서드
    for (int i = 0; i < dvd.size(); i++) {
        System.out.println("====[" + (i + 1) + "]====");
        dvd.get(i).showData();
    }
}
```

전체정보를 선택하면 현재 배열에 등록되어 있는 정보와 번호가 출력됩니다.

## 4. 기 능 소 개

# 대여 관리

## 대여 인터페이스

```
1 package rental;
2
3 public interface RentalInterface {
4
5     void borrow();
6     void returnBack();
7     void borrowInfo();
8
9 }
```

대여, 반납, 대여정보확인 기능을 구현해야한다.

## 대여 클래스 변수

```
16 public class Rental implements RentalInterface{
17
18     private HashMap<BookInfo,String[]> rentalBookInfo;
19     private HashMap<GameInfo,String[]> rentalGameInfo;
20     private HashMap<DvdInfo,String[]> rentalDVDInfo;
21
22     private AccountManager account = AccountManager.getInstance();
23     private GameManage game = GameManage.getInstance();
24     private DvdManager dvd = DvdManager.getInstance();
25     private BookManage book = BookManage.getInstance();
26
27     private int userindex;
28
29     private SimpleDateFormat format = new SimpleDateFormat ( "yyyy-MM-dd HH:mm:ss");
30
31     private Rental(){
32         rentalBookInfo = new HashMap<BookInfo,String[]>();
33         rentalGameInfo = new HashMap<GameInfo,String[]>();
34         rentalDVDInfo = new HashMap<DvdInfo,String[]>();
35     }
36
```

대여 정보를 담는 변수 : rentalBookInfo, rentalGameInfo, rentalDVDInfo  
계정 변수: account, 게임 변수: game, DVD 변수: dvd, 책 변수: book  
시간 포맷 변수: format, 로그인중인 사용자의 인덱스 정보: userindex



## 메뉴 기능

```
46● public int showMenu() {  
47     System.out.println("=====");  
48     System.out.println("메뉴를 선택해주세요.");  
49     System.out.println(Util.BORROW + ". 대여" + Util.RETURN + ". 반납" + Util.BORROWINFO + "대여 정보");  
50     System.out.println("=====");  
51  
52     int select = Util.keyboard.nextInt();  
53  
54     return select;  
55 }  
56  
57● public int showMenu2() {  
58     System.out.println("=====");  
59     System.out.println("카테고리를 선택해주세요.");  
60     System.out.println(Util.BOOK + ". 책" + Util.DVD + ". DVD" + Util.GAME + ". 게임");  
61     System.out.println("=====");  
62  
63     int select = Util.keyboard.nextInt();  
64  
65     return select;  
66 }
```

메뉴를 보여주는 메서드입니다.  
선택한 결과값을 int로 반환합니다.

## 대여 기능

```
68* @Override
69 public void borrow() {
70
71     int choice = showMenu2();
72
73     switch(choice) {
74
75     case Util.BOOK:
76         book.searchBook();
77         if(book.getSearchResult().size() < 1) {
78             System.out.println("검색 결과가 없습니다.");
79             return;
80         }
81
82         System.out.println("대여할 상품 번호를 입력하세요.");
83         int borrowSelect = Util.keyboard.nextInt();
84
85         BookInfo selected = book.getSearchResult().get(borrowSelect-1);
86
87         if(selected.getRental().equals("대여 가능")) {
88             selected.rental = "대여 중";
89
90             rentalBookInfo.put(selected, new String[]{ account.getLoginId() , format.format(new Date()) });
91
92             for(int i = 0; i < account.getMembers().size(); i++) {
93                 if(account.getMembers().get(i).getID().equals(account.getLoginId())){
94                     account.getMembers().get(i).addInfo(selected);
95                 }
96             }
97
98             System.out.println("대여 되었습니다.");
99
100
101         }else {
102             System.out.println("대여중 입니다.");
103             System.out.println("대여자: "+rentalGameInfo.get(selected)[0]+
104                 "대여일: "+rentalGameInfo.get(selected)[1]);
105         }
106         break;
```

메뉴2를 보여준 후 case문으로 책, 게임, DVD중 선택한 것을 대여하는 기능으로 넘어갑니다.

검색 기능으로 대여할 책을 검색하고 검색 결과중에 대여할 상품을 선택하면 대여가능한 상태면 대여 정보에 책 객체와 빌린 사용자 ID, 빌린 날짜를 저장합니다.

사용자객체에도 빌린 책 정보를 저장합니다.

대여중이면 대여자의 ID와 빌린 날짜를 보여줍니다.

## 대여 정보 확인 기능

```
221• @Override
222 public void borrowInfo() {
223     System.out.println("사용자가 대여중인 목록입니다.");
224
225     for(int i = 0; i < account.getMembers().size(); i++) {
226         if(account.getMembers().get(i).getID().equals(account.getLoginId())){
227             userindex = i;
228             for(int k = 0; k < account.getMembers().get(i).getinfo().size(); k++) {
229                 System.out.println("=====["+k+"]=====");
230                 account.getMembers().get(userindex).getinfo().get(k).toString();
231                 if(account.getMembers().get(userindex).getinfo().get(k) instanceof GameInfo) {
232                     System.out.println("대여자 ID: "+rentalGameInfo.get(account.getMembers().get(userindex).getinfo().get(k))[0]);
233                     System.out.println("대여일: "+rentalGameInfo.get(account.getMembers().get(userindex).getinfo().get(k))[1]);
234                 }else if(account.getMembers().get(userindex).getinfo().get(k) instanceof BookInfo) {
235                     System.out.println("대여자 ID: "+rentalBookInfo.get(account.getMembers().get(userindex).getinfo().get(k))[0]);
236                     System.out.println("대여일: "+rentalBookInfo.get(account.getMembers().get(userindex).getinfo().get(k))[1]);
237                 }else if(account.getMembers().get(userindex).getinfo().get(k) instanceof DvdInfo) {
238                     System.out.println("대여자 ID: "+rentalDVDInfo.get(account.getMembers().get(userindex).getinfo().get(k))[0]);
239                     System.out.println("대여일: "+rentalDVDInfo.get(account.getMembers().get(userindex).getinfo().get(k))[1]);
240                 }
241                 System.out.println("=====");
242             }
243         }
244     }
```

로그인한 사용자의 대여 정보를 보여주는 기능입니다.  
대여한 책, DVD, 게임의 정보와 대여자ID, 대여일을 보여줍니다.

## 반납 기능

```
173* @Override
174 public void returnBack() {
175
176     borrowInfo();
177
178     System.out.println("반납할 대상의 번호를 선택하세요. ");
179
180     int selectNumber = Util.keyboard.nextInt();
181
182     if(account.getMembers().get(userindex).getinfo().get(selectNumber) instanceof GameInfo) {
183
184         for(int i = 0; i < game.getGame().size(); i++) {
185             if(account.getMembers().get(userindex).getinfo().get(selectNumber).equals( game.getGame().get(i)) ) {
186                 game.getGame().get(i).rental = "대여 가능";
187             }
188         }
189         rentalGameInfo.remove(account.getMembers().get(userindex).getinfo().get(selectNumber));
190
191     }else if(account.getMembers().get(userindex).getinfo().get(selectNumber) instanceof BookInfo) {
192
193         for(int i = 0; i < book.getBook().size(); i++) {
194             if(account.getMembers().get(userindex).getinfo().get(selectNumber).equals( book.getBook().get(i)) ) {
195                 book.getBook().get(i).rental = "대여 가능";
196             }
197         }
198         rentalBookInfo.remove(account.getMembers().get(userindex).getinfo().get(selectNumber));
199
200     }else if(account.getMembers().get(userindex).getinfo().get(selectNumber) instanceof DvdInfo) {
201
202         for(int i = 0; i < dvd.getDvd().size(); i++) {
203             if(account.getMembers().get(userindex).getinfo().get(selectNumber).equals( dvd.getDvd().get(i)) ) {
204                 dvd.getDvd().get(i).rental = "대여 가능";
205             }
206         }
207
208         rentalDVDInfo.remove(account.getMembers().get(userindex).getinfo().get(selectNumber));
209     }
210
211     account.getMembers().get(userindex).getinfo().remove(selectNumber);
212
213     System.out.println("반납 되었습니다.");
214
215 }
```

로그인한 사용자가 자신이 대여한 목록을 검색하고 반납할 목록을 선택하면 사용자의 대여 정보에서 선택한 목록을 삭제하고 책,DVD,게임 객체에서 대여중인 상태를 대여 가능상태로 바꾼 후 대여 정보에서 반납한 목록을 삭제합니다.

## 메인 기능

```

10 public class RentalSystemMain {
11
12     public static void main(String[] args) {
13         AccountMain accountMain = new AccountMain();
14         BookMain bookMain = new BookMain();
15         DvdMain dvdMain = new DvdMain();
16         GameMain gameMain = new GameMain();
17         Rental rental = Rental.getInstance();
18
19         System.out.println("=====대여 시스템을 시작합니다.=====");
20
21         accountMain.accountMainRun1();
22         if(accountMain.manager.getLoginId().equals("admin")) {
23             System.out.println("=====관리자로 로그인 하였습니다.=====");
24             System.out.println("=====관리자용 시스템으로 접속합니다.=====");
25
26             while(true) {
27                 System.out.println("=====이용하실 메뉴를 선택하세요.=====");
28                 System.out.println(Util.BOOK+" ". 도서 관리Wn" + Util.DVD +" ". DVD 관리Wn" + Util.GAME
29                     + " ". 게임 관리Wn" + Util.USER + " ". 회원 관리Wn" +Util.QUIT +" ". 종료 );
30                 int choice = Util.keyboard.nextInt();
31                 if(choice == Util.BOOK) {
32                     System.out.println("=====도서 관리 시스템에 접속합니다.=====");
33                     bookMain.bookMainRun();
34                 }else if( choice == Util.DVD) {
35                     System.out.println("=====DVD 관리 시스템에 접속합니다.=====");
36                     dvdMain.dvdMainRun();
37                 }else if(choice == Util.GAME) {
38                     System.out.println("=====게임 관리 시스템에 접속합니다.=====");
39                     gameMain.gameMainRun();
40                 }else if(choice == Util.USER) {
41                     System.out.println("=====회원 관리 시스템에 접속합니다.=====");
42                     accountMain.accountMainRun2();
43                 }else if(choice == Util.QUIT) {
44                     System.out.println("=====시스템을 종료합니다.=====");
45                     break;
46                 }else {
47                     System.out.println("=====제대로 된 번호를 입력해 주세요.=====");
48                 }
49             }
50         }
51
52     }else {
53         System.out.println("=====사용자로 로그인 하였습니다.=====");
54         System.out.println("=====사용자용 시스템으로 접속합니다.=====");
55     }

```

## 1. 로그인

## 2-1.관리자용 시스템

2-1-1.도서관리 시스템

2-1-2.DVD관리 시스템

2-1-3.게임관리 시스템

2-1-4.회원관리 시스템

## 메인 기능

```

56 while(true) {
57     System.out.println("=====이용하실 메뉴를 선택하세요.=====");
58     System.out.println(Util.BORROW+" ".대여Wn" + Util.RETURN+" ".반납Wn" + Util.USERRETURN+
59         ". 개인 대여정보 조회Wn" + Util.SEARCHALL+" ". 도서/DVD/게임 검색Wn" +Util.QUIT+" ". 종료" );
60
61     int choice = Util.keyboard.nextInt();
62
63     if(choice == Util.BORROW) {
64         System.out.println("=====대여 시스템에 접속합니다.=====");
65         rental.borrow();
66     }else if( choice == Util.RETURN) {
67         System.out.println("=====반납 시스템에 접속합니다.=====");
68         rental.returnBack();
69     }else if(choice == Util.USERRETURN) {
70         System.out.println("=====개인 대여정보를 조회합니다.=====");
71         rental.borrowInfo();
72     }else if(choice == Util.SEARCHALL) {
73         System.out.println("=====도서/DVD/게임 검색 시스템에 접속합니다.=====");
74         System.out.println("=====검색하실 메뉴를 선택하세요.=====");
75         System.out.println(Util.BOOK+" ". 도서 검색Wn" + Util.DVD+" ". DVD 검색Wn" + Util.GAME+
76             ". 게임 검색Wn" +Util.QUIT+" ". 종료" );
77         choice = Util.keyboard.nextInt();
78
79         if(choice == Util.BOOK) {
80             System.out.println("=====도서를 검색합니다.=====");
81             bookMain.bookManage.searchBook();
82         }else if( choice == Util.DVD) {
83             System.out.println("=====DVD를 검색합니다.=====");
84             dvdMain.manager.searchDvd();
85         }else if(choice == Util.GAME) {
86             System.out.println("=====게임을 검색합니다.=====");
87             gameMain.gameManage.searchGame();
88         }else if( choice == Util.QUIT) {
89             System.out.println("=====검색을 종료합니다.=====");
90             break;
91         }else {
92             System.out.println("=====제대로 된 번호를 입력해 주세요.=====");
93         }
94
95     }else if(choice == Util.QUIT) {
96         System.out.println("=====시스템을 종료합니다.=====");
97         break;
98     }else {
99         System.out.println("=====제대로 된 번호를 입력해 주세요.=====");
100     }
101 }
102

```

## 2-2.사용자용 시스템

2-2-1.대여 시스템

2-2-2.반납 시스템

2-2-3.개인 대여정보 시스템

2-2-4.검색 시스템

2-2-4-1.도서 검색

2-2-4-2.DVD 검색

2-2-4-3.게임 검색

## 5. 문제해결 및 개선방향

## 구현중 발생했던 문제와 해결방안

문제	해결방안
예외처리	예외처리 메서드들을 적합한 형태로 수정
잘못된 접근제어자	접근제어자를 수정
보기 불편한 구조	클래스를 나눠서 메서드를 속성별로 정리
잘못된 자료구조 사용	자료구조 형태 변경
싱글톤에 대한 이해부족	팀원들의 피드백과 꾸준한 학습을 통한 이해



## 협업과정에서 발생한 문제

문제	해결방안
깃허브에 master 브랜치로 push해서 이전 버전이 섞임	로컬에서 전 버전을 보고 고쳐서 다시 push함
기능 이름이 통일되 있지 않아서 통합에 문제가 발생	이름을 명확히 숙지하고 통합

## 개선 방향

개선 사항	내용
예외처리	키보드 입력을 받을때 더 적절한 예외 처리를 하도록 개선해야 한다. 인덱스 에러가 나지 않도록 처리 해 주어야 한다.
코드 간결화	반복되는 코드를 for문을 통해 더 간결하게 줄인다.
파일 입출력	검색 결과를 명세서 형태로 파일로 저장 가능하게 한다.
인터페이스 통합	중복되는 기능을 가진 인터페이스들을 하나로 통합한다.
검색기능 향상	정규식을 이용한 부분일치하는 문자 검색이 가능하게 한다.

**THANK  
YOU**